

Monitorování aktivních síťových prvků pomocí webové služby

Monitoring of Active Network Elements Using Web Services

Zadání bakalářské práce

Student:

Zdeněk Nábělek

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2601R013 Telekomunikační technika

Téma:

Monitorování aktivních síťových prvků pomocí webové služby
Monitoring of Active Network Elements Using Web Services

Zásady pro vypracování:

1. Způsoby monitorování aktivních síťových prvků.
2. Naprogramování rozhraní pro monitorování aktivních síťových prvků.
3. Vytvoření webového rozhraní pro přístup k monitorovaným parametrům.
4. Testování aplikace v reálných podmínkách.

Seznam doporučené odborné literatury:


Dle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Libor Michalek, Ph.D.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



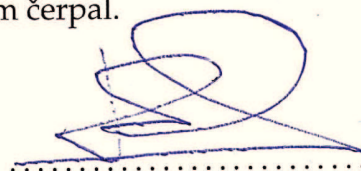
prof. RNDr. Vladimír Vašínek, CSc.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 5. května 2012

A handwritten signature in blue ink, consisting of several loops and a horizontal line at the bottom, positioned above a dotted line.

Chtěl bych poděkovat svému vedoucímu bakalářské práce, Ing. Liboru Michalkovi, Ph.D. za cenné rady a připomínky k mé práci. Dále bych chtěl poděkovat všem, kteří mě podporovali a pomohli mi umožnit tuto bakalářskou práci vypracovat.

Abstrakt

Bakalářská práce se zabývá základním popisem protokolu SNMP, návrhem a vytvořením aplikace pro monitorování aktivních síťových prvků. Aplikace získává hodnoty sledovaných parametrů a ukládá je do databáze MySQL. Zpracovaná data jsou pro uživatele interpretována pomocí webové aplikace, která byla vytvořena v jazyce PHP za použití Javascriptu a kaskádových stylů.

Klíčová slova: SNMP, monitorování, aktivní síťové prvky, PHP

Abstract

The bachelor's thesis deals with the basic description of the SNMP protocol, the design and creation of applications for monitoring of active network elements. The application obtains the values of monitored parameters and stores them in MySQL database. The processed data are interpreted for the user through a web application which was created in PHP using Javascript and CSS.

Keywords: SNMP, monitoring, active network elements, PHP

Seznam použitých zkratk a symbolů

BW	– Bandwidth - šířka pásma
CCS	– Cascading Style Sheets - kaskádové styly
DB	– Database - databáze
HTML	– HyperText Markup Language - označovací jazyk pro hypertext
HTTP	– HyperText Transfer Protocol - hypertextový přenosový protokol
ICMP	– Internet Control Message Protocol - internetový protokol řídicích zpráv
IF	– InterFace - rozhraní
IT	– Information Technology - informační technologie
MIB	– Management Information Base - informační báze dat pro správu
OID	– Object Identifier - objekt identifikátor
PHP	– Hypertext Preprocessor - hypertextový preprocesor
SNMP	– Simple Network Management Protocol - jednoduchý protokol správy sítě
Syslog	– Standard for computer data logging - standard pro záznam programových zpráv
UDP	– User Datagram Protocol - uživatelský datagramový protokol
UPS	– Uninterruptible Power Supply - nepřerušitelný zdroj napájení
WMI	– Windows Management Instrumentation - nástrojové vybavení pro správu Windows
WWW	– World Wide Web - celosvětová síť (pavučina) distribuovaný soubor dokumentů v Internetu, lze jej hypertextově prohledávat

Obsah

1	Úvod	4
2	Způsoby monitorování aktivních síťových prvků	6
2.1	ICMP	6
2.2	Syslog	6
2.3	SNMP	6
3	Naprogramování rozhraní pro monitorování aktivních síťových prvků	11
3.1	Vytvoření databáze	11
3.2	Skripty obsluhující výčet hodnot	11
3.3	Skripty pro připojení k databázi	14
3.4	Skript pro zaznamenávání hodnot do grafu	15
4	Vytvoření webového rozhraní pro přístup k monitorovaným parametrům	16
4.1	Úvod	16
4.2	Seznam prvků	16
4.3	Administrace	19
4.4	Grafy	20
4.5	Informace	23
4.6	Přihlášení - odhlášení	25
5	Testování aplikace v reálných podmínkách	26
6	Závěr	28
7	Reference	29
	Přílohy	29
A	Ukázka skriptu zajišťujícího výčet hodnot	30

Seznam obrázků

1	Ukázka příkazu ping pod Linuxem	7
2	Ukázka výpisu syslogu z PC stanice s Linuxem	7
3	Ukázka výpisu SNMP GET z PC stanice s Linuxem	8
4	Ukázka výpisu SNMP GETNEXT z PC stanice s Linuxem	9
5	Ukázka výpisu SNMP BULKGET z PC stanice s Linuxem	9
6	Ukázka výpisu SNMP WALK z PC stanice s Linuxem	9
7	Ukázka výpisu SNMP GET z routeru Cisco 2621	10
8	Schéma	12
9	Datový model databáze MySQL	12
10	Vytvoření databáze pomocí SQL skriptu	13
11	Vytvoření databáze pomocí PHPMyAdmin	14
12	Úvodní obrazovka	17
13	Seznam prvků	18
14	Administrační rozhraní pro vkládání údajů o aktivních prvcích . .	19
15	Graf 1	21
16	Graf 2	22
17	Informace a kontaktní formulář	24
18	Ukázka programu Getif	27

Seznam výpisů zdrojového kódu

1	Ukázka spuštění skriptu v crontabu	13
2	Funkce pro připojení k databázi	14
3	Funkce pro odpojení od databáze	14
4	Výpis proměnné COUNTER ze souboru funkce.php	15
5	Výpis seznamu na základě přihlašovacích údajů	17
6	Selekt z databáze pro výpis do seznamu	18
7	Ukázka části skriptu zpracovávajícího vložené údaje	20
8	Ukázka části Javascriptu nastavujícího barvu grafu	21
9	Část skriptu odesílajícího emailem informace zadané uživatelem .	23
10	Funkce hlídající přihlášení uživatele	25

1 Úvod

Základním předpokladem zajištění provozuschopnosti a dosažení požadované dostupnosti technických systémů je nepřetržité monitorování jejich stavu. V systémech většího rozsahu nelze sledování zajistit lidskými silami, proto je nutné tuto činnost automatizovat.

V této práci se pozornost zaměřuje na oblast IT technologií speciálně počítačové sítě. Jak vývoj počítačových sítí, tak vývoj monitorování síťových prvků, jde neustále kupředu a má mnoho podob. Monitorováním se zabývá spousta firem, které nabízí různé způsoby jeho implementace a řešení. Jedná se v zásadě o sběr, analýzu a vyhodnocení provozních parametrů sledovaných zařízení. Na začátku je důležité si rozmyslet, jaká oblast má být předmětem monitorování (servery, provoz nebo aktivní síťové prvky), jaké výstupy budou požadovány a jaké budou použity prostředky. Dle způsobu zavedení lze monitorování rozdělit do tří skupin:

- Vytvoření vlastního dohledového systému
- Freeware řešení tzv. Open source
- Využití komerčního řešení

K samotnému napsání kódu je zapotřebí dobrá znalost programovacích jazyků, čas pro vývoj a minimální náklady. Výsledkem je program zhotovený na míru požadavkům pro monitorování firemního prostředí.

Mezi nejznámější open source programy patří Nagios, Catci a Zabbix. Tento software je v "surovém" stavu. K jeho funkčnosti je nutné přizpůsobit a doprogramovat části kódu, což není vždy jednoduché a vyžaduje značnou časovou náročnost.

Komerční řešení nabízí relativně snadné a rychlé nasazení do provozu při minimalizaci požadavku na vlastní lidské zdroje. Tyto programy jsou většinou nainstalovány "na pár kliknutí" a po minimálním nastavení připraveny k použití, avšak často bývají finančně nákladné. Mezi nejrozšířenější patří HP OpenView, CiscoWorks LAN Management Solution, dále pak Zenoss, PacketTrap pt360 a další.

Tématem bakalářské práce je monitorování aktivních síťových prvků, které je realizováno naprogramováním vlastního kódu. Bakalářská práce je rozdělena do několika kapitol. V druhé kapitole je popsáno monitorování, možné způsoby dohledu a stručný popis základních testů. Ve třetí kapitole je uveden typ zvolené databáze, možnosti jejího vytvoření a shrnutí jednotlivých skriptů obsluhujících získávání hodnot ze síťových prvků. Čtvrtá kapitola je zaměřena na vytvoření rozhraní pro koncového uživatele. Pátá kapitola je věnována zavedení aplikace do datové sítě a otestování v reálných podmínkách.

Programovací kód byl z převážné části napsán v jazyce PHP, který vznikl v roce 1996 a zkratka znamenala Personal Home Page. Od té doby jazyk PHP prošel řadou velkých změn a nyní tato zkratka znamená Hypertext Preprocessor. PHP je programovací jazyk pracující na straně serveru. Pomocí něj lze ukládat, měnit a mazat data WWW stránek. PHP skript se nejprve provede na serveru a prohlížeči je odeslán pouze výsledek. PHP není nikterak složitý k pochopení a už se základy si běžný uživatel vystačí.

Dále byl při programování použit Javascript. Jedná se o skriptovací jazyk, který umožňuje vytvořit hodiny, hodnotit data ve formuláři, počítat, dynamizovat data, umožňuje tvorbu všemožných prvků k oživení webu, od blikajících textů až po jednoduché hry. Javascript je základem dynamického webu. Pro práci s Javascriptem je dobré mít základní znalosti HTML. Javascript je závislý na prohlížeči a je na uživateli, zda bude povolen nebo zakázán. Na rozdíl od PHP, zdrojový kód Javascriptu a HTML je možno zobrazit. Další rozdíl oproti PHP je v tom, že Javascript se vykonává přímo v prohlížeči [1].

Grafická podoba aplikace byla realizována s užitím CSS, tj. Cascading Style Sheets, což se do češtiny překládá jako kaskádové styly. Jsou to kolekce metod pro grafickou úpravu webových stránek. Kaskádové proto, že se na sebe mohou vrstvit definice stylů, ale platí jenom ta poslední. CSS může být použito následujícími třemi způsoby:

- Přímo v textu zdroje u formátovaného elementu pomocí atributu `style="..."`. Tomu se říká přímý styl. Je to nešikovné, ale občas se s tím setkáváme.
- Pomocí stylopisu (angl. stylesheet) v hlavičce stránky. Stylopis je seznam stylů, v němž je přesně nadefinován vzhled stránky. V kódu se stylopis píše mezi tagy `<style>` a `</style>`.
- Použitím externího stylopisu – to je soubor `*.css`, na který se stránka odkazuje tagem `<link>` a je v něm umístěný stylopis. Hlavní výhodou je, že na jeden takový soubor se dá napárovat mnoho stránek, takže pak všechny vypadají podobně.

Není nutné ovládat všechny tři způsoby aplikace CSS. Nejčastěji používaným způsobem je externí soubor [9].

Pro ukládání získaných hodnot byla zvolena databáze MySQL. Databáze se dá snadno nainstalovat na různé operační systémy. V současné době patří mezi nejvíce používané databáze, nejen s ohledem na snadnou instalaci a výkon, ale také proto, že se jedná o volně šiřitelný software. Velmi oblíbená a často užívaná je kombinace Linux, MySQL, PHP a Apache jako základní software webového serveru („technologie LAMP“) [6].

2 Způsoby monitorování aktivních síťových prvků

2.1 ICMP

Velmi jednoduchým a v praxi často využívaným způsobem testování dostupnosti síťových prvků je využití protokolu ICMP, zejména příkazu PING. Výstupem je výpis (viz.: obr. 1), který nás informuje o dostupnosti zařízení, době odezvy, tj. době, kdy paket urazil cestu k cíli a zpět. Na konci výpisu je celkové vyhodnocení, ze kterého lze vyčíst úspěšnost, ztrátovost, minimální, maximální a průměrnou dobu odezvy. Tento protokol má jen omezené možnosti, a proto se většinou kombinuje ještě s jinými způsoby monitorování.

2.2 Syslog

Do češtiny by se slovo syslog dalo volně přeložit jako standard pro zaznamenávání události ze serveru. Pod tímto pojmem si většinou představíme nějaký soubor, kde se ukládají různé informace o chodu, stavu, pádech nebo chybách běhu serveru (viz.: obr. 2). Tímto jeho použitím ale zdaleka nekončí. Může být také nakonfigurován jako syslog server, který bude zaznamenávat a zpracovávat přijímané zprávy z jiných zařízení. Na nich je obdobně nainstalován klient, který zprostředkovává odesílání logovaných událostí na syslog server.

2.3 SNMP

SNMP je jednoduchý, široce rozšířený a užitečný standardizovaný protokol, který slouží k získávání nebo nastavování hodnot na určitém zařízení. Obdobou je například WMI od firmy Microsoft. Výhodou oproti syslogu je bezpochyby jeho vicestranné využití. Jedná se o získávání konkrétních údajů na základě jednoznačného OID, o výpis několika hodnot, ale také o nastavování hodnot na zařízení. S ohledem na jednotnou databázi MIB, která obsahuje jednotlivé OID, je oblast použitelnosti značně otevřená a rozšířená. Pomocí tohoto protokolu máme k dispozici rozšířené možnosti reakce na zjištěný stav kontrolovaného zařízení.

Monitorování pomocí protokolu SNMP bylo vybráno pro bakalářskou práci vzhledem k široké oblasti jeho využití, univerzálnosti a vlastnostem. Níže jsou uvedeny základní vlastnosti SNMP protokolu [2]:

- Model klient - server
- Široká škála podpory ze strany zařízení
- Verze
- Rychlost

- SNMP dotazy
- Trap
- Znalostní databáze MIB

```

zdarec@zdarec-G550 ~ $ ping www.seznam.cz -c 5
PING www.seznam.cz (77.75.72.3) 56(84) bytes of data.
64 bytes from www.seznam.cz (77.75.72.3): icmp_req=1 ttl=248 time=11.6 ms
64 bytes from www.seznam.cz (77.75.72.3): icmp_req=2 ttl=248 time=11.8 ms
64 bytes from www.seznam.cz (77.75.72.3): icmp_req=3 ttl=248 time=10.7 ms
64 bytes from www.seznam.cz (77.75.72.3): icmp_req=4 ttl=248 time=11.2 ms
64 bytes from www.seznam.cz (77.75.72.3): icmp_req=5 ttl=248 time=11.5 ms

--- www.seznam.cz ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 10.774/11.405/11.816/0.387 ms
zdarec@zdarec-G550 ~ $ █

```

Obrázek 1: Ukázka příkazu ping pod Linuxem

```

zdarec@zdarec-G550 ~ $ cat /var/log/syslog | grep 11:20:36
Apr 22 11:20:36 zdarec-G550 NetworkManager[928]: <info> (eth1): device state change: activated -> unavailable (reason 'none') [100 20 0]
Apr 22 11:20:36 zdarec-G550 NetworkManager[928]: <info> (eth1): deactivating device (reason 'none') [0]
Apr 22 11:20:36 zdarec-G550 NetworkManager[928]: <info> (eth1): canceled DHCP transaction, DHCP client pid 1795
Apr 22 11:20:36 zdarec-G550 avahi-daemon[931]: Withdrawing address record for 192.168.2.102 on eth1.
Apr 22 11:20:36 zdarec-G550 avahi-daemon[931]: Leaving mDNS multicast group on interface eth1.IPv4 with address 192.168.2.102.
Apr 22 11:20:36 zdarec-G550 avahi-daemon[931]: Interface eth1.IPv4 no longer relevant for mDNS.
Apr 22 11:20:36 zdarec-G550 NetworkManager[928]: <info> Policy set 'Wired connection 1' (eth0) as default for IPv4 routing and DNS.
Apr 22 11:20:36 zdarec-G550 NetworkManager[928]: <info> Policy set 'Wired connection 1' (eth0) as default for IPv4 routing and DNS.
Apr 22 11:20:36 zdarec-G550 wpa_supplicant[959]: CTRL-EVENT-DISCONNECTED bssid=00:00:00:00:00:00 reason=0
Apr 22 11:20:36 zdarec-G550 NetworkManager[928]: <info> (eth1): taking down device.
Apr 22 11:20:36 zdarec-G550 avahi-daemon[931]: Interface eth1.IPv6 no longer relevant for mDNS.
Apr 22 11:20:36 zdarec-G550 avahi-daemon[931]: Leaving mDNS multicast group on interface eth1.IPv6 with address fe80::c2cb:38ff:fe31:b591.
Apr 22 11:20:36 zdarec-G550 avahi-daemon[931]: Withdrawing address record for fe80::c2cb:38ff:fe31:b591 on eth1.
Apr 22 11:20:36 zdarec-G550 dbus[918]: [system] Activating service name='org.freedesktop.nm_dispatcher' (using servicehelper)
Apr 22 11:20:36 zdarec-G550 NetworkManager[928]: <info> WiFi hardware radio set disabled
Apr 22 11:20:36 zdarec-G550 NetworkManager[928]: <info> WiFi now disabled by radio killswitch
Apr 22 11:20:36 zdarec-G550 dbus[918]: [system] Successfully activated service 'org.freedesktop.nm_dispatcher'
zdarec@zdarec-G550 ~ $ █

```

Obrázek 2: Ukázka výpisu syslogu z PC stanice s Linuxem

Model klient - server

Protokol SNMP vyžaduje pro komunikaci dvě strany. Jednou entitou je serverová část a druhou je klientská část. SNMP pracuje ve dvou režimech činnosti:

- Klienti posílají dotazy na server a přijímají odpovědi. Dotazy mohou být uskutečňovány v různém časovém rozmezí.
- Zařízení zasílá oznámení (trap) v předem definovaných situacích (například překročení nastavené hodnoty - prahu) na adresu serveru pro příjem zpráv.

Široká škála podpory ze strany zařízení

Podporu SNMP má velká řada zařízení, například aktivní síťové prvky, počítačové komponenty, tiskárny, teplotní a vlhkostní čidla, UPS, klimatizace. Dále pak speciální zařízení jako například: přepěťové, proudové ochrany, jističe a mnoho dalších.

Verze

Protokol SNMP nyní existuje ve třech verzích. SNMPv1 a SNMPv2c používají pro autentizaci community string, v podstatě textové heslo. V SNMPv3 je možno využít autentizaci pomocí jména, hesla a šifrování.

Rychlost

SNMP používá pro komunikaci UDP protokol, díky jeho vlastnostem je komunikace velmi rychlá, ale může docházet ke ztrátě (nedoručení) zasílané informace (paketu). Proto je od verze 2 implementována kontrola doručení, takže by ke ztrátě nemělo dojít. Standardně se používá port 161 (SNMP) na straně agenta (pro dotazy) a port 162 (SNMPTRAP) na straně serveru (pro trapy). Klient, který posílá dotaz, zvolí dynamický port, z kterého posílá dotaz na port 161. Agent odpovídá z portu 161 na dynamický port klienta. V praxi je pro každý dotaz použit jiný dynamický port.

SNMP dotazy

Klasická komunikace, kdy se dotaz posílá na jednu hodnotu a následně je přijímána odpověď, probíhá následovně:

- odešle se dotaz – nastaví se typ GET, zadá se OID pro zjišťovanou hodnotu, vlastní hodnota se nastaví na NULL
- vrátí se odpověď – typ je nastaven na RESPONSE (2), OID na dotazovanou hodnotu a je vyplněna hodnota

Možné typy dotazů jsou SNMP GET – vrátí jednu hodnotu (viz.: obr. 3), GET-NEXT - vrátí další hodnotu (viz.: obr. 4) (vezme následující OID za zadaným). Od verze 2 GET-BULK (viz.: obr. 5), který vrací více hodnot najednou (například pro všechny porty přepínače). SNMP WALK vrací strom od zadané pozice dle OID (viz.: obr. 6). Pomocí SNMP můžeme hodnoty nejen číst, ale také zapisovat, k tomu slouží typ SET.

```
zdarec@zdarec-G550 ~ $ snmpget -v2c -c public 127.0.0.1 iso.3.6.1.2.1.1.5.0
iso.3.6.1.2.1.1.5.0 = STRING: "zdarec-G550"
zdarec@zdarec-G550 ~ $
```

Obrázek 3: Ukázka výpisu SNMP GET z PC stanice s Linuxem

```
zdarec@zdarec-G550 ~ $ snmpgetnext -v2c -c public 127.0.0.1 iso.3.6.1.2.1.1.4.0
iso.3.6.1.2.1.1.5.0 = STRING: "zdarec-G550"
zdarec@zdarec-G550 ~ $
```

Obrázek 4: Ukázka výpisu SNMP GETNEXT z PC stanice s Linuxem

```
zdarec@zdarec-G550 ~ $ snmpbulkget -v2c -c public 127.0.0.1 iso.3.6.1.2.1.1.0
iso.3.6.1.2.1.1.1.0 = STRING: "Linux zdarec-G550 3.0.0-12-generic #20-Ubuntu SMP
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (480380) 1:20:03.80
iso.3.6.1.2.1.1.4.0 = STRING: "Me <me@example.org>"
iso.3.6.1.2.1.1.5.0 = STRING: "zdarec-G550"
iso.3.6.1.2.1.1.6.0 = STRING: "Sitting on the Dock of the Bay"
iso.3.6.1.2.1.1.7.0 = INTEGER: 72
iso.3.6.1.2.1.1.8.0 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.11.3.1.1
zdarec@zdarec-G550 ~ $
```

Obrázek 5: Ukázka výpisu SNMP BULKGET z PC stanice s Linuxem

```
zdarec@zdarec-G550 ~ $ snmpwalk -v2c -c public 127.0.0.1 iso.3.6.1.2.1.1.9
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The SNMP Management Architecture MIB."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The MIB for Message Processing and Dispatching."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The management information definitions for the SNMP
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "The MIB module for managing TCP implementations"
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing IP and ICMP implementati
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing UDP implementations"
iso.3.6.1.2.1.1.9.1.3.8 = STRING: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.4.1 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.2 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.3 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.5 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.6 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.7 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.8 = Timeticks: (4) 0:00:00.04
zdarec@zdarec-G550 ~ $
```

Obrázek 6: Ukázka výpisu SNMP WALK z PC stanice s Linuxem

Informační báze dat pro správu

Každá hodnota v MIB databázi je jednoznačně identifikována číselným identifikátorem OID, který je tvořen posloupností čísel oddělených tečkou. Daná hodnota vzniká z OID nadřazeného prvku doplněním tečky a aktuálního čísla pozice. Celá tato stromová struktura je uložena v MIB databázi. Navíc MIB databáze obsahuje jména a popisy jednotlivých hodnot (OID). MIB databáze může být doplněna o další hodnoty pomocí části struktury uložené v MIB souboru.

Ukázkou OID může být například hodnota .1.3.6.1.2.1.2.2.1.6.1 (viz.: obr. 7), které odpovídá textová verze z MIB databáze:

iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifPhysAddress

```
dohled-CVS:~ # snmpget -v2c -c public 10.68.45.254 1.3.6.1.2.1.2.2.1.6.1
IF-MIB::ifPhysAddress.1 = STRING: 0:8:e3:fb:13:40
dohled-CVS:~ #
```

Obrázek 7: Ukázka výpisu SNMP GET z routeru Cisco 2621

3 Naprogramování rozhraní pro monitorování aktivních síťových prvků

Navržený a realizovaný dohledový systém umožňuje získávání hodnot parametrů z aktivních síťových prvků, jejich zpracování a interpretaci (viz.: obr. 8). Je tvořen třemi základními moduly:

- Vyčítání hodnot parametrů sledovaných zařízení
- Uložení načtených dat do databáze
- Prezentace výsledků

3.1 Vytvoření databáze

Pro databázi bylo vybráno prostředí MySQL, které patří v kombinaci s PHP mezi nejpoužívanější. Aby byla zaručena funkčnost aplikace a bylo možno pracovat se všemi daty, které je zapotřebí ukládat a mít k dispozici, byl navržen následující model databáze (viz.: obr. 9).

Vlastní vytvoření databáze může být realizováno dvěma způsoby:

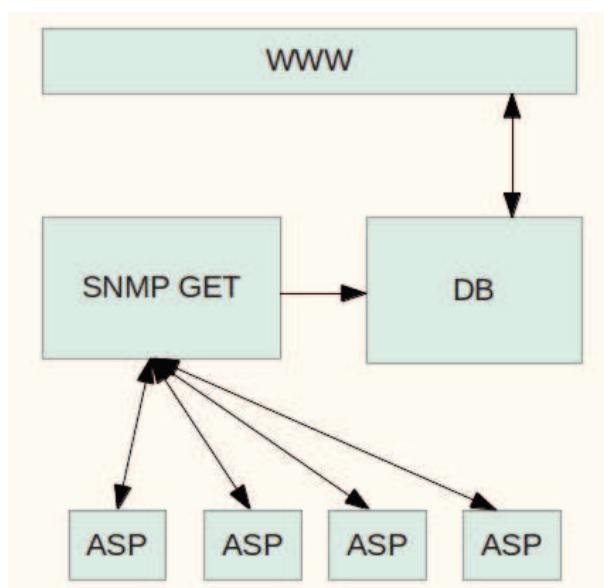
1. Pomocí SQL skriptu (viz.: obr. 10)
2. Interaktivně pomocí webového rozhraní PHPMyAdmin (viz.: obr. 11)

Tímto způsobem vytvořená databáze je připravena k použití. Po vygenerování databáze je nezbytně nutné nastavit příslušná uživatelská oprávnění.

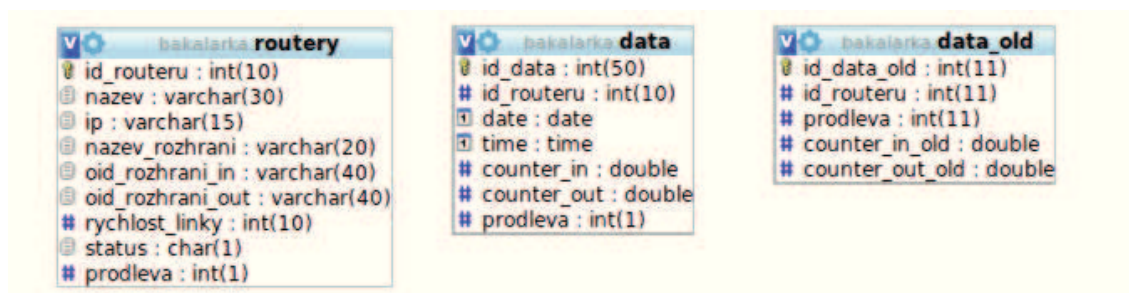
3.2 Skripty obsluhující výčet hodnot

Nejčastější použití je měření objemu datového toku s využitím zjišťování hodnot čítačů přenesených bajtů. Z praxe vyplynulo, že je zapotřebí mít více skriptů, respektive umět je spouštět v různých časových intervalech, a to z důvodu přesnosti a prioritizace měření. Proto jsou napsány celkem čtyři skripty umístěné v adresáři *výčet* (viz.: příloha CD):

- snmp_req0.php pro interval 30 sekund
- snmp_req1.php pro interval 1 minutu
- snmp_req2.php pro interval 5 minut
- snmp_req3.php pro interval 10 minut



Obrázek 8: Schéma



Obrázek 9: Datový model databáze MySQL

Jednoduchou úpravou mohou být přidány další časové intervaly. Výše zmíněné skripty plně pokrývají běžné požadavky, jelikož z praktického hlediska jsou delší časové intervaly pro vyhodnocování irelevantní. S prodlužující se periodou měření se snižuje míra přesnosti (detailu). To znamená, že celkový objem přenesených dat za dobu odpovídající periodě měření je reprezentován jedinou průměrnou hodnotou a dochází ke ztrátě informací o skutečném časovém průběhu datového toku.

Princip obsluhy je jednoduchý. Jsou-li web a databáze umístěny například na linuxovém systému, pak je v crontabu (viz.: výpis 1) zajištěno spouštění skriptů v požadovaných časech. V databázi je pro každý měřený prvek uložena informace o periodě měření. Na základě tohoto údaje jsou vždy vybrány prvky, u nichž se v daném čase provádí měření. Například ve skriptu spouštěném s periodou

```

CREATE TABLE `data` (
  `id_data` int(50) NOT NULL auto_increment,
  `id_routeru` int(10) default NULL,
  `date` date default NULL,
  `time` time default NULL,
  `counter_in` double default NULL,
  `counter_out` double default NULL,
  `prodleva` int(1) default NULL,
  PRIMARY KEY (`id_data`),
  KEY `date_time` (`date`,`time`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci AUTO_INCREMENT=1;

CREATE TABLE `data_old` (
  `id_data_old` int(11) NOT NULL auto_increment,
  `id_routeru` int(11) default NULL,
  `prodleva` int(11) default NULL,
  `counter_in_old` double default NULL,
  `counter_out_old` double default NULL,
  PRIMARY KEY (`id_data_old`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci AUTO_INCREMENT=1;

CREATE TABLE `routery` (
  `id_routeru` int(10) NOT NULL auto_increment,
  `nazev` varchar(30) collate utf8_czech_ci NOT NULL,
  `ip` varchar(15) collate utf8_czech_ci NOT NULL,
  `nazev_rozhrani` varchar(20) collate utf8_czech_ci NOT NULL,
  `oid_rozhrani_in` varchar(40) collate utf8_czech_ci NOT NULL,
  `oid_rozhrani_out` varchar(40) collate utf8_czech_ci NOT NULL,
  `rychlost_linky` int(10) NOT NULL,
  `status` char(1) collate utf8_czech_ci NOT NULL,
  `prodleva` int(1) NOT NULL,
  PRIMARY KEY (`id_routeru`),
  KEY `status` USING BTREE (`status`,`prodleva`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci AUTO_INCREMENT=1;

```

Obrázek 10: Vytvoření databáze pomocí SQL skriptu

jedna minuta jsou získávány informace ze všech zařízení, která mají nastavenou hodnotu "prodleva" na 1. Přesměrování v tomto případě do `/dev/null` zajišťuje, že výstup skriptu se nebude vypisovat na obrazovku [7, 4, 3].

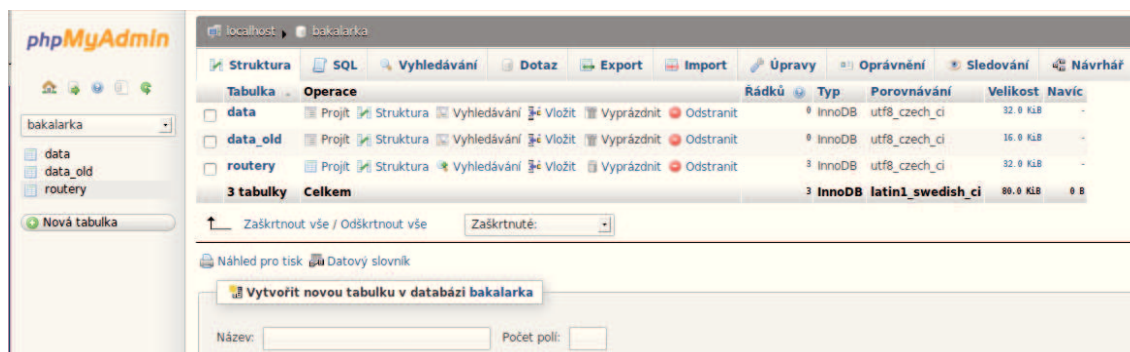
```

*/1 * * * * /usr/bin/php /srv/www/htdocs/mrtg/snmp_req0.php > /dev/null 2>&1
*/1 * * * * /usr/bin/php /srv/www/htdocs/mrtg/snmp_req1.php > /dev/null 2>&1
*/5 * * * * /usr/bin/php /srv/www/htdocs/mrtg/snmp_req2.php > /dev/null 2>&1
*/10 * * * * /usr/bin/php /srv/www/htdocs/mrtg/snmp_req3.php > /dev/null 2>&1

```

Výpis 1: Ukázka spuštění skriptu v crontabu

Následně se provede SNMP dotaz, přepoččet a získaná hodnota se uloží do databáze. Informace k provedení dotazu (IP adresa, OID, na které se má provést



Obrázek 11: Vytvoření databáze pomocí PHPMyAdmin

dotaz) jsou zadávány při vkládání dat nového objektu k měření. Community string je definován na začátku skriptu.

3.3 Skripty pro připojení k databázi

Aby bylo možné pracovat s daty, je zapotřebí se nejprve k databázi připojit, poté provést požadovanou operaci a následně řádně ukončit spojení s databází. K tomu jsou naprogramovány funkce `db_open` (viz.: výpis 2) a `db_close` (viz.: výpis 3) [5].

```
function db_open()
{
    global $spojeni;
    @ $spojeni = MySQL_connect(DB_POCITAC, DB_JMENO, DB_HESLO) or die("<h3>
        Nepodařilo se připojit k databázi</h3>");
    MySQL_query("SET CHARACTER SET utf8");
    @ MySQL_select_db(DB_DATABAZE) or die("<h3>Nepodařilo se vybrat databázi</
        h3>");
}
```

Výpis 2: Funkce pro připojení k databázi

```
function db_close()
{
    global $spojeni;
    return MySQL_Close($spojeni);
}
```

Výpis 3: Funkce pro odpojení od databáze

Proměnné *DB_POCITAC*, *DB_JMENO* a *DB_HESLO* jsou uloženy ve zvláštním souboru *pristup_do_db.php* (viz.: Příloha CD) z důvodu jednoduché přenositelnosti a snazší práce s databází.

3.4 Skript pro zaznamenávání hodnot do grafu

Skript pro zaznamenávání hodnot do grafu je rovněž součástí již zmíněných skriptů v adresáři *vycet*. Pracuje na následujícím principu (viz.: Přílohy):

1. Získání hodnoty čítače
2. Naformatování na celé číslo
3. Přetypování hodnoty
4. Získání předchozí hodnoty pro výpočet
5. Zapsání předchozí hodnoty pro další měření
6. Zapsání nové hodnoty, která bude sloužit při příštím měření jako předchozí hodnota
7. Výpočet naměřeného rozdílu
8. Zapsání konvertované hodnoty pro vykreslení grafu

V dnešní době máme k dispozici několik způsobů, jak lze hodnoty zpracovávat a zaznamenávat do grafů. Grafy jsou vykreslovány pomocí Javascriptu a knihovny *FLOT*R [11, 10].

Aby během procesu nedocházelo ke zkreslování údajů, je nutné zmínit definovanou proměnnou *COUNTER* (viz.: výpis 4) v souboru *funkce.php*, kterou volají skripty z adresáře *vycet*. Tato proměnná představuje maximální hodnotu čítače, tj. 2^{32} , na aktivních prvcích od firmy CISCO. Je-li hodnota 2^{32} překročena, začíná se počítat od nuly a proměnná *COUNTER* začíná hrát svou roli.

```
define("COUNTER", 4294967296);
```

Výpis 4: Výpis proměnné *COUNTER* ze souboru *funkce.php*

4 Vytvoření webového rozhraní pro přístup k monitorovaným parametrům

Stránky byly naprogramovány tak, aby práce s nimi byla co nejjednodušší a nejeftivnější. Menu a ukončení stránky je vkládáno ze souboru *begin.php* a *end.php* (viz.: příložené CD). Díky tomu je i úprava menu možná na jednom místě, ale projeví se na všech stránkách. Současně se také při načítání kontroluje, zda je uživatel přihlášen, či nikoliv.

4.1 Úvod

Úvodní stránka je přihlašovací a obsahuje základní informace k bakalářské práci (viz.: obr. 12). Na ukázkou je přihlášení řešeno pro dva uživatele:

- Administrátor (admin/admin)
- Běžný uživatel (test/test)

Po přihlášení jsou načteny do session údaje o přihlášeném uživateli, na jejichž základě probíhá zpřístupnění ostatních částí webové aplikace. V praxi jsou běžní uživatelé ověřováni oproti firemní databázi. Tím zajistíme přístup všem, kteří ho potřebují. Ale jen uživatelé s administrátorským právem mají možnost editace, vkládání, mazání, spouštění či zastavování měření. Vše je v souboru *index.php*.

4.2 Seznam prvků

Seznam má dvě podoby. Jedna je pro administrátora, druhá pro běžného uživatele. Kliknutí na Seznam z menu se načte stránka *seznam.php*. Zároveň dochází k ověřování aktivního uživatele a provede se výběr ze tří možností (viz.: výpis 5):

- Je přihlášen administrátor - dochází k přesměrování na *seznamadmin.php* (viz.: obr. 13)
- Je přihlášen běžný uživatel - dochází k přesměrování na *seznamuser.php*
- Přihlášení nebylo provedeno - načte se hlášení o chybě a kontaktování administrátora



Obrázek 12: Úvodní obrazovka

```

<?php
include 'begin.php';
if (isset($_SESSION['auth_adm']))
{
    // Je přihlasen administrator
    include 'seznamadmin.php';
}
else
{
    if (isset($_SESSION['auth_test']))
    {
        // Je přihlasen uzivatel
        include 'seznamusr.php';
    }
    else
    {
        // nikdo neni přihlasen
        include 'error.php';
    }
}
}

```



```
include 'end.php';
?>
```

Výpis 5: Výpis seznamu na základě přihlašovacích údajů

The screenshot shows a web application titled "Monitorování aktivních síťových prvků". It has a navigation bar with links: Index, Seznam, Vložit údaje, Grafy, About, and Odhlásit. Below the navigation bar is a section titled "Seznam měřených uzlů". This section contains a table with the following data:

Název routeru	Rozhraní	IP adresa	Rychlost	Měřit	Status měření	Vymazat	Editovat
ag-kmbr	ser1/0	10.60.80.2	128				
@ISDN 1	CPU ISDN	10.34.255.135	0				
@ISDN 2	CPU ISDN	10.34.255.136	0				
@test	CPU vakvi	10.34.255.131	0				
ag-bv	ser1/0	10.56.104.2	1024				
ag-d1p503010	Serial0/3/0	10.53.18.2	64				
ag-d5p513102	Serial0/3/0	10.53.197.2	64				
ag-dvjm	Serial1/2:0	10.56.64.2	512				
ag-dvsc	Gi0/0	10.224.124.242	4096				
ag-dvsc	ser0/2	10.48.64.2	2048				
ag-dvsm	ser1/0	10.60.64.2	256				
ag-dvsm-d	ser1/0	10.60.200.2	128				
ag-dvvc	Gi0/0	10.224.160.18	2048				
ag-dvzc	ser0/3	10.44.64.2	128				
ag-fm	ser0/3	10.60.112.2	128				

Obrázek 13: Seznam prvků

Obdobně probíhá kontrola přihlášení na každé stránce. Celý obsah tohoto php skriptu je možno najít v souborech *seznam.php*, *seznamadmin.php*, *seznamuser.php*. Výpis je prezentován v přehledné tabulce na základě výsledku SQL dotazu (viz.: výpis 6).

```
db_open();
// Vyber parametry routeru z databaze – ulozi se do pole $row[0]
$query = @MySQL_Query("SELECT _nazev,nazev_rozhrani,ip,status,id_routeru,
    rychlost_linky,_FROM_routery_ORDER_BY_nazev_ASC,_nazev_rozhrani_ASC;");
$pocet_polozek = MySQL_num_rows($query);
db_close();
```

Výpis 6: Selekt z databáze pro výpis do seznamu

4.3 Administrace

Administrace je nejdůležitější pro funkčnost celé aplikace. Zadáávají se zde parametry měřeného aktivního prvku (viz.: obr. 14). V bakalařské práci bylo toto rozhraní naprogramováno tak, aby bylo nutno zadat dvě OID, které reprezentují čítače IN a OUT na straně IF. Dále je vyžadována IP adresa zařízení, rychlost měřeného IF reprezentována předem definovaným seznamem rychlostí a doplňující informace. Po odeslání formuláře je provedena kontrola zadaných údajů IN a OUT. Pokud není uživatelem vložena potřebná délka (hloubka) OID, je automaticky nastavena na nejvíce používané typy výčtových OID (viz.: výpis 7). Dále se provede připojení k databázi, vložení údajů, odpojení od databáze a přesměrování na stránku se seznamem.

The screenshot shows a web application titled "Monitorování aktivních síťových prvků". The navigation bar includes links: Index, Seznam, Vložit údaje, Grafy, About, and Odhlásit. The main content area is titled "Zadejte údaje pro identifikaci měřícího rozhraní." and contains a form with the following fields:

- Název uzlu(routeru):
- IP adresa uzlu:
- Název měřícího rozhraní:
- OID_IN měřícího rozhraní:
- OID_OUT měřícího rozhraní:
- Rychlost linky (Kbit/s):

At the bottom of the form is a button labeled "Uložit údaje do databáze".

Obrázek 14: Administrační rozhraní pro vkládání údajů o aktivních prvcích

```
// Predani hodnot
$nazev_uzlu      = $_POST['uzel'];
$ip_adresa      = $_POST['ip_adresa'];
$nazev_rozhrani  = $_POST['nazev_rozhrani'];

$nazev_rozhrani_in = $_POST['oid_in'];
if ( strlen($nazev_rozhrani_in) < 4 )
{
    $nazev_rozhrani_in = OID_IN . $nazev_rozhrani_in;
}

$nazev_rozhrani_out = $_POST['oid_out'];
if ( strlen($nazev_rozhrani_out) < 4 )
{
    $nazev_rozhrani_out = OID_OUT . $nazev_rozhrani_out;
}

$rychlost        = $_POST['rychlost'];
```

Výpis 7: Ukázka části skriptu zpracovávajícího vložené údaje

4.4 Grafy

Vykreslení grafu je rozděleno do dvou částí. V první části se volí nastavení zobrazení grafu tj. údaje požadované k jeho zobrazení (viz.: obr. 15). Druhá část je věnována samotnému vykreslení grafu s možností volby upřesňujících parametrů (viz.: obr. 16). Jedním z nich je nastavení dvouhodinových časových intervalů v rámci dne. Také osa rychlosti je flexibilní v rozmezí od 64Kbps do 4Gbps. Graf je vykreslován pomocí Javascriptu a doplňkového pluginu *flotr*. Dále se Javascript podílí na barevném provedení grafu (viz.: výpis 8) s nastavením pro odchozí datový provoz (upload) na modrou barvu a pro příchozí datový provoz (download) jasně zelenou. Výsledek měření je uživateli zobrazen co nejpřehledněji.

Celý tento proces můžeme shrnout do několika kroků následovně:

1. Uživatel vybere ze seznamu aktivní prvek, pro který chce graf vykreslit
2. Z databáze se provede select a vytáhnou se požadovaná data
3. Vygenerování dat pro grafy
4. Nastavení parametrů pro vykreslení
5. Vykreslení grafu

Monitorování aktivních síťových prvků

[Index](#)
[Seznam](#)
[+ Vložit údaje](#)
[Grafy](#)
[About](#)
[Odhlásit](#)

Nastavení zobrazení grafu

Vyberte uzel, na kterém chcete zobrazit vytížení linky:

ag-kmbr : ser1/0 : id.284

Nastavte datum ve formátu YYYY-mm-dd:

2012-03-26

[Přejít na zobrazení grafu](#)

Pozn.: pro přesné zobrazení grafu na následující stránce, vyberte u zobrazeného grafu správnou rychlost linky !

Obrázek 15: Graf 1

```

$( 'container' ),
[
  {
    data:
      [ <?php
        print $data_in;
      ?> ],
    label: "UPLOAD", color: 'blue',
    lines:
      {
        fill : false,
        show: true
      }
  },

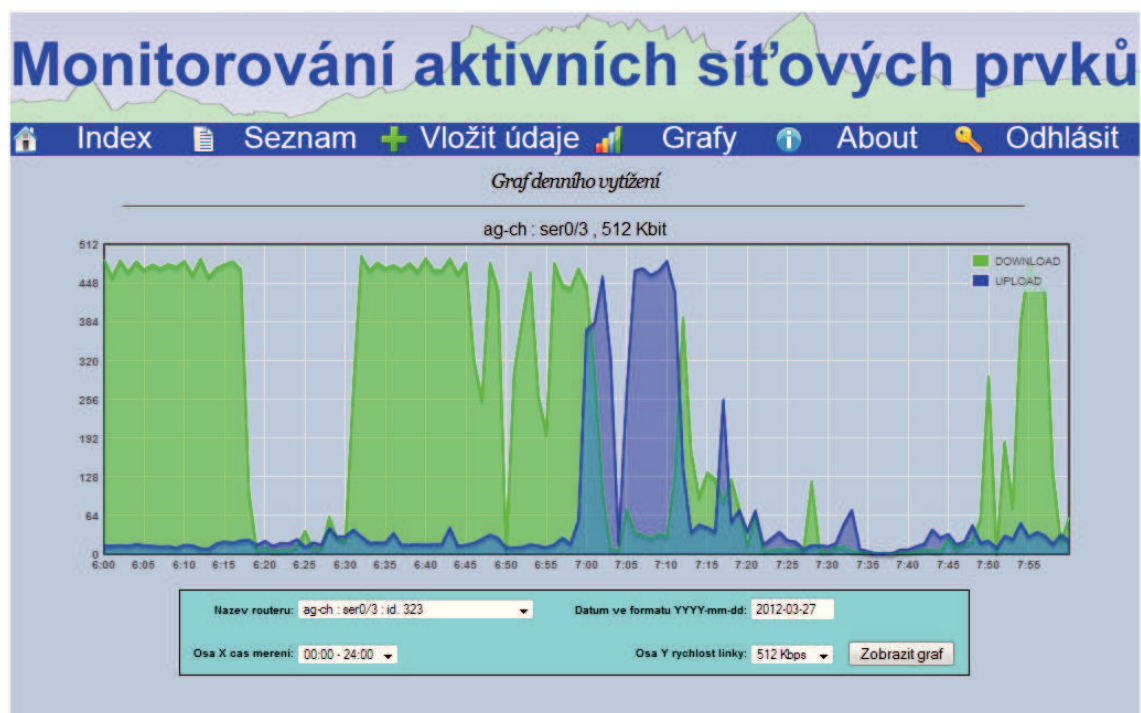
```

```

{
  data:
    [<?php
      print $data_out;
    ?>],
    label: "DOWNLOAD", color: '#02EE02',
    lines:
    {
      show: true, fill : true
    }
  },
],

```

Výpis 8: Ukázka části Javascriptu nastavujícího barvu grafu



Obrázek 16: Graf 2

4.5 Informace

Informace o autorovi jsou k dispozici na stránce s názvem *about.php* (viz.: obr. 17). Jedná se pouze o doplňující stránku, která není vázána na přihlášeného uživatele a obsahuje kontaktní formulář (viz.: výpis 9). Po jeho vyplnění jsou data (dotazy, připomínky, žádosti o povolení přístupu a podněty k zlepšení, ...) odeslána na předdefinovaný email.

```
<?php
if (ISSET($_POST["odeslat"]))
{
    $jmeno    = $_POST["jmeno"];
    $predmet  = $_POST["predmet"];
    $kontakt  = $_POST["kontakt"];
    $vzkaz    = $_POST["vzkaz"];
    $headers  = "Content-type:_text/plain;_charset=utf-8\n";
    $headers .= 'From:nab0008_bakalarka@dotazy.cz';
    $souhrn   = ("Jmeno:_$jmeno" . "Kontakt:_$kontakt" . "Vzkaz:_$vzkaz");
    if ($jmeno != "" && $predmet != "" && $kontakt != "" && $vzkaz != "")
    {
        mail("zdenek.nabelek.st@vsb.cz", $predmet, "
*****
Jmeno:_$jmeno_Kontakt:_$kontakt
*****

$vzkaz
*****",
        $headers);
        echo '<h4>Požadavek byl úspěšně odeslán</h4>';
    }
    else
        echo '<h3>Chyba! Zadejte všechny údaje. Formulář s dotazem nebyl odeslán.</h3>';
}
if (ISSET($_POST["zpet"]))
{
    header('Location:_index.php');
}
?>
```

Výpis 9: Část skriptu odesílajícího emailem informace zadané uživatelem

Monitorování aktivních síťových prvků

Index Seznam Vložit údaje Grafy About Odhlásit

Zdeněk Nábělek

Tel.: +420 777 05 71 45

@: zdenek.nabelek.st@vsb.cz

WWW: <http://home1.vsb.cz/~nab0008/>

Máte dotaz...? Volejte, pište e-mail, nebo odešlete kontaktní formulář

Jméno :

Předmět :

Kontakt / e-mail :

Vzkaz :

Odeslat Vynulovat Zpět na úvod

Obrázek 17: Informace a kontaktní formulář

4.6 Přihlášení - odhlášení

V tomto případě se nedá ani tak hovořit o webové stránce jako spíše o části menu respektive tlačítka (viz.: výpis 10) - část kódu, která nám umožní provést přihlášení uživatele nebo odhlášení aktivního uživatele. Vše je řešeno pomocí session, která se po přihlášení nastaví na hodnoty aktivního uživatele a následně je uchovává do doby, než dojde k jejímu zrušení tedy odhlášení uživatele. Aktivní session je využívána na všech stránkách, které jsou dostupné jen přihlášeným uživatelům.

```
<td valign="middle" align="center">
  <?php
    if (isset($_SESSION['auth_adm']) || isset($_SESSION['auth_test']))
    {
      ?>
      <a href="index.php?logout=yes" class="menu_logof"></a>
    <?php
    }
    else
    {
      ?>
      <a href="index.php" class="menu_login"></a>
    <?php
    }
  ?>
</td>
```

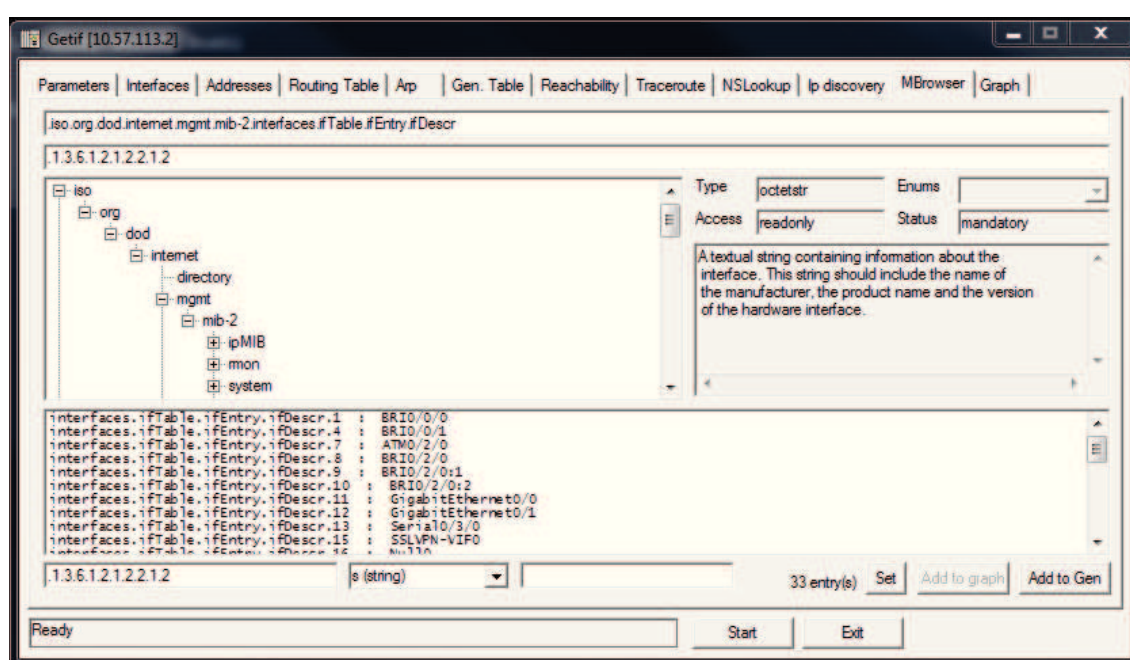
Výpis 10: Funkce hlídající přihlášení uživatele

5 Testování aplikace v reálných podmínkách

Kompletní zavedení a otestování proběhlo v rámci České pošty s.p., sekce provoz ICT. Toto oddělení se zabývá dohledem datové sítě ČP. Databáze i WWW aplikace byly nainstalovány na server s operačním systémem Linux SuSe. Do databáze bylo vloženo cca 150 koncových zařízení od firmy CISCO různých modelových řad.

Nejvíce využívanou skupinou bylo zaznamenávání dat v minutovém a pětiminutovém intervalu. Desetiminutový interval byl zapnut jen pro odzkoušení. Naopak v případě nárustu vytížení linky se následně zapínaly intervaly 30-ti sekundové pro zpřesnění měření. Vyhodnocení díky přehlednému grafu, obsahujícímu tok dat nejen do uzlu ale i ven, bylo rychlé a dostačující pro identifikaci problému (zda je linka vytížena více, než je normální stav). V případech, kdy se jednoznačně nedala určit příčina vytížení linek i přes to, že nárůst oproti normálnímu provozu byl znatelný, se zapínal druhý typ měření - NetFlow. S použitím NetFlow lze přesně určit zdrojovou i cílovou IP adresu a přesný objem toku dat mezi zdrojovou a cílovou stanicí.

OID na koncových prvcích je zjišťováno pomocí několika nástrojů pracujících na stejném principu. V tomto konkrétním případě byly hodnoty OID získávány za pomoci programu Getif (viz.: obr. 18), který obsahuje informace o nejrozšířenějších znalostních databázích MIB. Rychlost přípojných linek je známa a již uložena v existující databázi (zadaná administrátorem). Pokud tomu tak není, je možné rychlost získat buď z programu Getif, nebo vypsaním požadovaného IF po připojení na router, kde bude uveden záznam BW nebo rychlost IF.



Obrázek 18: Ukázka programu Getif

6 Závěr

Vytvořené prostředí je dobře adaptivní na různá rozhraní aktivních prvků. Jeho modifikaci lze využít k hlídání jiných čítačů jako jsou teplota, vlhkost, využití procesorů atd. Nejsme omezeni jedním výrobcem, ale použití je možné všude tam, kde je povoleno vyčítání hodnot pomocí protokolu SNMP. Napsané funkce a skripty ulehčí práci při "hlídání" provozních parametrů. Vzhledem k členitosti programu do několika souborů je používání velmi snadné a intuitivní. Při změně databáze dochází také ke změně přístupových parametrů. Díky koncepci můžeme tento zásah provést pouze v jednom souboru *pristup_do_db.php* a to nám dává k dispozici nové úložiště pro data.

Díky oddělenému rozhraní pro uživatele a administrátory jsou výsledky dostupné všem, kteří mají přihlašovací právo. Uvnitř podnikové sítě je většinou přihlášení nasměrováno na interní databázi uživatelů (např. LDAP). Běžní uživatelé mají právo zobrazení výstupních grafů, ale jen administrátor má možnost vkládat nové údaje, nebo provádět jejich změnu.

Tato aplikace má velký přínos v reálném provozu. Během chvíle lze na požádání zjistit stav vytížení jednotlivých měřených rozhraní. Díky přehlednému grafu je tento výstup snadno použitelný také pro report v inkriminovaných časech. V rámci firemní sítě se ujala a používá se.

Praxe ukázala, že aplikace splnila všechny požadavky a očekávání. Stávající aplikace by mohla být dále rozšířena například o emailový způsob notifikace při překročení nastavené prahové hodnoty, nebo vykreslení grafu za období zadané uživatelem (např. dva týdny).

Textová podoba bakalářské práce byla sepsána pomocí šablony Diploma v prostředí L^AT_EX[8].

7 Reference

- [1] Tvorba-webu. *Tvorba webu* [online]. 2003 - 2008 [cit. 2012-04-12]. Dostupné z: <http://www.tvorba-webu.cz/>
- [2] BOUŠKA, Petr. *Samuraj-cz.com* [online]. 2006 [cit. 2012-03-18].
- [3] HORÁK, Jaroslav. *Počítačové sítě pro začínající správce*. 4. aktualiz. a rozšiř. vyd. Praha: Computer Press, 2008, 327 s. ISBN 978-80-251-2073-6.
- [4] KRETCHMAR, James M. *Administrace a diagnostika sítí: pomocí OpenSource utilit a nástrojů*. 1. vyd. Brno: Computer Press, 2004, 216 s. ISBN 80-251-0345-5.
- [5] LECKY-THOMPSON, Ed a Steven D NOWICKI. *PHP 6: programujeme profesionálně*. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2010, 718 s. Programujeme profesionálně. ISBN 978-80-251-3127-5 (Váz.).
- [6] MASLAKOWSKI, Mark. *Naučte se MySQL za 21 dní*. 1. vyd. Praha: Computer Press, 2001, 478 s. ISBN 80-722-6448-6.
- [7] NEMETH, Evi, Garth SNYDER a Trent R HEIN. *Linux: kompletní příručka administrátora : 2. aktualizované vydání*. Vyd. 1. Brno: Computer Press, 2008, 984 s. ISBN 978-80-251-2410-9 (Váz.).
- [8] RYBIČKA, Jiří. *Latex pro začátečníky*. Brno: Konvoj, 2003, 238 s. ISBN 80-730-2049-1.
- [9] STANÍČEK, Petr. *Kompletní průvodce CSS: kaskádové styly*. Vyd. 1. Brno: Computer Press, 2003, 178 s. ISBN 80-722-6872-4.
- [10] ŠKULTÉTY, Rastislav. *JavaScript: programujeme internetové aplikace*. Vyd. 1. Praha: Computer Press, 2004, 224 s. ISBN 80-251-0144-4.
- [11] ŠKULTÉTY, Rastislav. *JavaScript: Programujeme internetové aplikace. Výuka jazyka do základů. Praktické ukázky využití. Podrobný popis objektů, vlastností a metod*. 1. vyd. Praha: Computer Press, 2001, 208 s. ISBN 80-722-6457-5.

A Ukázka skriptu zajišťujícího výčet hodnot

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD_HTML_4.01_Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Bakalářská práce Zdeněk Nábělek, NAB0008, KS</title>
    <link rel="icon" type="image/gif" href=" ../ pictures / favismall .png"/>
  </head>
  <body>
    <?php
      // Definice přístupového hesla
      define("COMMUNITY", "public");

      // Nactení knihoven
      require "../ funkce.php";

      // Podívám se do databaze, který router je zapnutý na měření a vytáhnu si o něm
      informace
      db_open();

      $stmt = "SELECT _id_routeru, _ip, _oid_rozhrani_in, _oid_rozhrani_out, _prodleva FROM _
        routery WHERE _status='A' AND _prodleva='2'";
      if (( $result = mysql_query($stmt, $GLOBALS['spojeni'])) )
      {
        while(($row = mysql_fetch_array($result, MYSQL_NUM)))
        {
          // Pomocné proměnné pro přehled
          $id_routeru = $row[0];
          $ipadresa   = $row[1];
          $oid_in     = $row[2];
          $oid_out    = $row[3];
          $prodleva   = $row[4];

          // Provedu snmp dotaz na dané routery
          if (!$snmpin = snmpget($ipadresa, COMMUNITY, trim($oid_in))) $snmpin = "0";
          if (!$snmpout = snmpget($ipadresa, COMMUNITY, trim($oid_out))) $snmpout
            = "0";

          // Naformátování proměnné na číselné
          $snmpin_hodnota = trim(substr($snmpin, strpos($snmpin, ".")+1));
          $snmpout_hodnota = trim(substr($snmpout, strpos($snmpout, ".")+1));

          // Přetypování hodnoty
          settype($snmpin_hodnota, "double");
          settype($snmpout_hodnota, "double");

          // Zjistím čas a datum

```

```

$date = date("Y-m-d");
$time = date("H:i:s");

// Zapsani predchozi hodnoty pro dalsi mereni
$query = "SELECT counter_in_old, counter_out_old FROM data_old WHERE
        id_routeru='$id_routeru' AND prodleva='2'";
$sovereni = @mysql_query ($query);

if (mysql_num_rows($sovereni) == 0) // Je volno, zacina uplne nove mereni
{
    $query = "INSERT INTO data_old (id_routeru, prodleva, counter_in_old,
        counter_out_old) VALUES ('$id_routeru', '1', '$snmpin_hodnota',
        '$snmpout_hodnota')";
    $vysledek = @mysql_query($query);
    if ($vysledek)
    {
        // V poradku data jsou ulozena, presmerovani na dalsi stranku
        echo '<br><font.size="2" color="green">Záznam uložen do databáze
            _tabulka_data_old</font><br>';
        $snmpin_hodnota_old = $snmpin_hodnota;
        $snmpout_hodnota_old = $snmpout_hodnota;
    }
    // Nastala chyba
    else
    {
        echo '<br><font.size="2" color="red">Chyba při zápisu do databáze
            _tabulka_data_old</font><br>';
    }
}
else
{
    // Ziskani predchozi hodnoty – v databazi uz je udaj
    $query = @MySQL_Query("SELECT counter_in_old, counter_out_old FROM
        data_old WHERE id_routeru='$id_routeru' AND prodleva='2'");
    while ($row = MySQL_Fetch_Row($query))
    {
        $snmpin_hodnota_old = $row[0];
        $snmpout_hodnota_old = $row[1];
    }
    // Zapisi novou hodnotu, která bude sloužit při prístim mereni jako
    predchozi
    $query = @MySQL_Query("UPDATE data_old SET 'counter_in_old'='
        $snmpin_hodnota', 'counter_out_old'='$snmpout_hodnota' WHERE
        id_routeru='$id_routeru' AND prodleva='2'");
}

// Vypocet namereneho rozdilu
if ( $snmpin_hodnota < $snmpin_hodnota_old )
{

```

```

        $snmpin_hodnota = COUNTER - $snmpin_hodnota_old + $snmpin_hodnota;
    }
    else
    {
        $snmpin_hodnota = $snmpin_hodnota - $snmpin_hodnota_old;
    }

    if ( $snmpout_hodnota < $snmpout_hodnota_old )
    {
        $snmpout_hodnota = COUNTER - $snmpout_hodnota_old +
            $snmpout_hodnota;
    }
    else
    {
        $snmpout_hodnota = $snmpout_hodnota - $snmpout_hodnota_old;
    }

    // Prepočet na Kbit / s
    //      počet prenesených byte / minuta
    // : 30      počet prenesených byte / sekundu
    // * 8      počet prenesených bitu / sekundu
    // : 1000   počet prenesených Kbitu / sekundu
    $snmpin_hodnota = $snmpin_hodnota / 30 * 8 / 1024;
    $snmpout_hodnota = $snmpout_hodnota / 30 * 8 / 1024;

    // Zapisu do databaze namerene udaje + ID_INTERVALU pro 1 minutu = 1
    $query = "INSERT INTO data_(id_routeru, date, time, counter_in, counter_out,
        prodleva) VALUES ('$id_routeru', '$date', '$time', '$snmpin_hodnota',
        '$snmpout_hodnota', '1');";
    $vysledek = @mysql_query($query);
    if ($vysledek)
    {
        // V poradku data jsou ulozena, presmerovani na dalsi stranku
        echo '<br><font.size="2" color="green">Záznam uložen do databáze _ _
            tabulka_data</font><br>';
    }
    // Nastala chyba
    else echo '<br><font.size="2" color="red">Chyba při zápisu do databáze _ _
        tabulka_data</font><br>';
    }
}

db_close();

?>
</body>
</html>

```

Výpis 11: Ukázka skriptu zajišťujícího výčet hodnot